

國立臺灣大學電機資訊學院電信工程學研究所

碩士論文

Graduate Institute of Communication Engineering

College of Electronic Engineering & Computer Science

National Taiwan University

Master Thesis

用於全景影像卷積操作之編碼機制

Omnidirectional Image Encoding

楊宗山

Tsung-Shan Yang

指導教授: 吳沛遠 博士

Advisor: Pei-Yuan Wu Ph.D.

中華民國 110 年 7 月

July, 2021

國立臺灣大學碩士學位論文

口試委員會審定書



用於全景影像卷積操作之編碼機制

Omnidirectional Image Encoding

本論文係楊宗山君 (R08942065) 在國立臺灣大學電信工程學研究所完成之碩士學位論文，於民國 110 年 7 月 5 日承下列考試委員審查通過及口試及格，特此證明

吳沛遠 *Pei-Yuan*

口試委員：\_\_\_\_\_

(指導教授)

丁建均

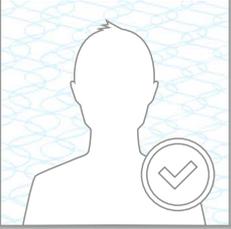
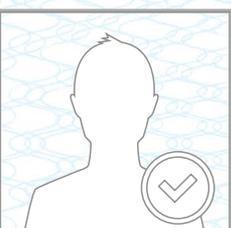
王鈺強

所長：\_\_\_\_\_ *吳沛遠*

# Signature Certificate

Document Ref.: YJTNZ-8Q4KK-MAUZA-2JW9X

Document signed by:

	<p><b>王鈺強 Yu-Chiang</b> Verified E-mail: ycwang@ntu.edu.tw</p> <p>IP: 140.112.41.145      Date: 05 Jul 2021 07:08:53 UTC</p>	<p>王鈺強</p> 
	<p><b>丁建均</b> Verified E-mail: jjding@ntu.edu.tw</p> <p>IP: 140.112.238.146      Date: 05 Jul 2021 07:12:50 UTC</p>	<p>丁建均</p> 
	<p><b>吳沛遠 Pei-Yuan</b> Verified E-mail: peiyuanwu@ntu.edu.tw</p> <p>IP: 123.193.33.72      Date: 05 Jul 2021 08:50:16 UTC</p>	<p>吳沛遠 <i>Pei-Yuan</i></p> 

Document completed by all parties on:

05 Jul 2021 08:50:16 UTC

Page 1 of 1



Signed with PandaDoc.com

PandaDoc is a document workflow and certified eSignature solution trusted by 25,000+ companies worldwide.







# Acknowledgements

碩班兩年的時間，讓我理解讀書和做研究的差異。釐清問題本質、找出改進之處、嚴謹的實驗、寫論文的邏輯，讓我成長許多，也使我在未來的道路有更多的選擇。

首先感謝吳沛遠教授，無論是做研究的態度、計畫的實作、投影片的製作和報告的邏輯等都給予我扎實的訓練。論文的撰寫也給予我很多實用的意見和改進。

感謝盧建宏博士，讓我體驗美國的文化，並和我討論論文的方向和進度，也提供我未來方向上許多寶貴的意見。

感謝博理 530 實驗室所有的同學。柏偉、宗憲、博閔、汝晉學長帶我融入實驗室和習慣研究生的生活。子毅、達軒、緯濬、芃苒和我交換學業上的意見，也是一同努力畢業的好夥伴，一起替博理 530 注入歡樂的氣氛。

也感謝台大競技啦啦隊，在我研究之餘提供一個讓我消遣的地方，在社團的日子經歷了很多事，也一起完成在大專盃的第一面金牌，付出的汗水讓我在人生的道路上除了研究也有一段難忘的回憶。

最後謝謝我的家人，在我的碩班期間給我很多自由，讓我不用擔心經濟上的壓力，也不會干涉我的選擇，在我遇到困難時會和我討論，支持我完成我的碩士論文。





## 摘要

本篇基於編碼機制，以編碼的方式改變作用於平面影像的卷積核中的權重，使卷積在全景影像的特徵提取上能有較佳的表現，並且可以與現存的卷積類神經網路模塊兼容。

實驗結果以全景圖片分類的準卻度呈現了此編碼機制和卷積類神經網路及殘差模塊的相容性，並以 omni-MNIST, omni-CIFAR10, omni-CIFAR100 進行實驗，在準確度上得到目前最佳的結果。

**關鍵字：**全景影像、卷積、編碼機制





# Abstract

An encoding mechanism for omni-directional images is proposed. It extends convolution widely used in planar images to omni-directional images on sphere surface. Unlike other convolution kernels, the proposed mechanism gives an add-on solution to be adapted in the existing neural network architectures to process omni-directional images. Experimental results on convolution neural network (CNNs) and residual nets demonstrate the effectiveness of the proposed mechanism, achieving state-of-the-art performance in omni-MNIST, omni-CIFAR10, and omni-CIFAR100 datasets.

**Keywords:** Omnidirectional, Encoding, Convolution

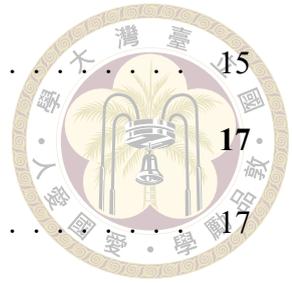




# Contents

	<b>Page</b>
<b>Verification Letter from the Oral Examination Committee</b>	<b>i</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>摘要</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>Contents</b>	<b>ix</b>
<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xiii</b>
<b>Chapter 1 Introduction</b>	<b>1</b>
<b>Chapter 2 Related Work</b>	<b>5</b>
2.1 CNNs Based on Different Projections . . . . .	5
2.2 Spherical CNNs . . . . .	6
2.3 Self-defined Representations and Kernels . . . . .	7
<b>Chapter 3 Method</b>	<b>9</b>
3.1 Convolution and Self-Attention on Feature Extraction . . . . .	10
3.2 Encoding with Positional Information . . . . .	11
3.3 Spherical absolute encoding . . . . .	12
3.4 Spherical relative encoding . . . . .	13

3.5	Encoding and Convolution . . . . .	15
<b>Chapter 4</b>	<b>Experiment</b>	<b>17</b>
4.1	Dataset . . . . .	17
4.2	Classification on omniMNIST . . . . .	17
4.2.1	Experiment Setup . . . . .	18
4.2.2	Results . . . . .	19
4.3	Residual Module . . . . .	19
4.3.1	Experiment Setup . . . . .	20
4.3.2	Result . . . . .	20
4.4	Self-Attention Model . . . . .	21
4.4.1	Result . . . . .	22
<b>Chapter 5</b>	<b>Conclusion</b>	<b>25</b>
	<b>References</b>	<b>27</b>





# List of Figures

1.1	Distortion near poles (red part), discontinuity on two sides (blue part), and normal grid near equator (yellow part) . . . . .	2
3.1	In the $3 \times 3$ kernel size setting, the pixels which are further away from the center (blue) represent greater surface area, and is assigned more attention score through <b>Spherical encoding</b> . (eq. 3.12) . . . . .	16
3.2	Process of <b>Spherical encoding</b> (eq. 3.12). Input can be either omnidirectional images or feature maps from previous layers. . . . .	16
4.1	Omnidirectional dataset includes images with severe distortion around the North and South poles and discontinuity on the left and right sides. . . . .	17
4.2	The Resnet-18 backbone. . . . .	20
4.3	The physical meaning of pixel on equirectangular projection and on SphereNet kernel. (a) The pixel on the actual sphere surface by equirectangular projection (b) The receptive field of SphereNet kernel. . . . .	22





# List of Tables

4.1	Classification accuracy comparison of the proposed spherical encoding and various baselines on omni-MNIST. . . . .	18
4.2	Classification accuracy on multi-layered models of various depths on omni-MNIST. . . . .	18
4.3	Comparison of various convolution/encoding schemes on ResNet18. . . . .	20
4.4	Comparison between spherical encoding, convolution, and SphereNet . . . . .	22
4.5	results for different encodings on omni-cifar10 . . . . .	23
4.6	Comparison of various encoding schemes on omni-cifar100 . . . . .	23





# Chapter 1 Introduction

Omnidirectional image gives a panoramic representation of physical spaces when compared to regular 2D images. It enables significant applications such as Google Street View, virtual tours of real estate, 360° showcase of automobile, and virtual-reality experience. Nowadays, omnidirectional image can be acquired by either consumer electronics such as smartphones or 360° cameras, or professional devices such as 3D scanner[20] or DSLR cameras with a rotator.

Spherical signals are often represented as a variety of data types, such as image grid[12], point clouds[23], or voxels[21][26]. Equirectangular projection is perhaps the most prevailing way to project various spherical signals onto the 2D space. Equirectangular projection is used extensively in image capturing, while most of the omnidirectional image datasets are composed of equirectangular projected images.

Equirectangular projection maps the longitude and latitude coordinates of the spherical image to the x- and y-coordinates of a plane image, respectively. This, however, is often accompanied with undesired distortions around the north and south poles, as well as discontinuity on the two sides (see fig. 4.1). As a result, special care must be taken before applying conventional euclidean-based learning algorithms [3] to equirectangular images projected from spherical signals.

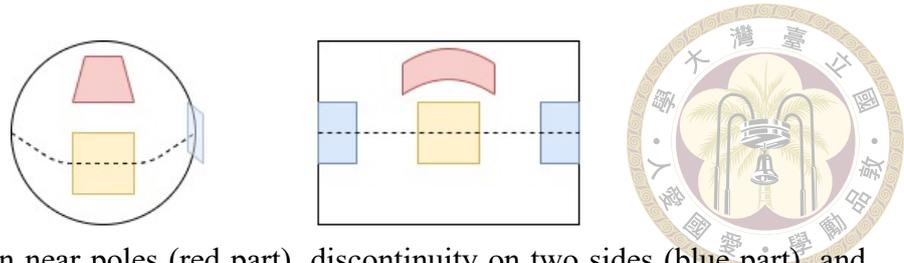


Figure 1.1: Distortion near poles (red part), discontinuity on two sides (blue part), and normal grid near equator (yellow part)

To deal with the distortion and discontinuity issues as mentioned above, multiple learning schemes were proposed such as Spherical CNN[3] and SphereNet[6]. In spherical CNN, the mathematical framework of rotation-invariant convolution on  $SO(3)$  is derived and analyzed. By regarding the unit sphere  $S^2$  as the quotient map  $SO(3)/SO(2)$ , a spherical signal can be extended to a function on  $SO(3)$ , on which spherical CNN can be applied. However, it is still not clear how the spherical CNN framework can be applied to other prevailing modules beyond convolutions. Furthermore, since the convolution is performed on the three-dimensional  $SO(3)$  manifold instead of the conventional two-dimensional  $S^2$  manifold, the memory usage becomes a dire burden.[1]

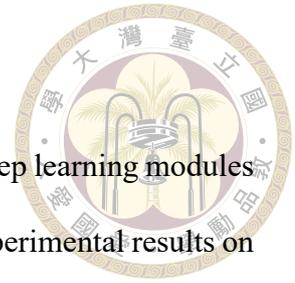
SphereNet kernel[4] retains the receptive field on the tangent plane and computes the offset of the sampling locations of the convolution filters based on gnomonic projection. The reprojection local region addresses the heavy distortion issue of patterns near the North and South poles and the uniform sphere sampling addresses the issue of over-weighting on high-latitude regions. However, the inconsistency of physical meaning the extracted feature and the equirectangular projected image makes the kernel incompatible to residual modules. (see Sec. 4.3 for more details)

In this work, we propose a spherical encoding to improve the performance of convolution neural network on omnidirectional data. The contributions are listed as below:

- We propose a spherical encoding based on great circle distance to calibrate the con-

volution weights on distorted regions at high-latitudes.

- The proposed spherical encoding is compatible to prevailing deep learning modules such as residual module, with its effectiveness supported by experimental results on omnidirectional image classification tasks.
- We give a physical interpretation of each pixel computed from different kinds of convolution on equirectangular projected images in the aspect of receptive fields.







## Chapter 2 Related Work

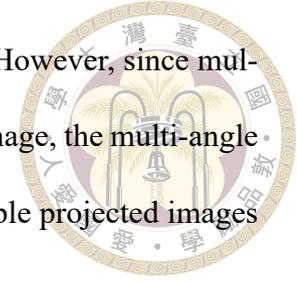
Many convolution-based models have been proposed for image processing tasks due to the great success of convolution neural network (CNN). However, conventional convolution is originally designed for planar images, which may not be ideal for spherical images where distortion and discontinuity issues must be reckoned with (see fig. 4.1).

### 2.1 CNNs Based on Different Projections

An intuitive solution to solve the distortion problem in equirectangular projected image is to refine the distorted areas through multiple perspective projections. Lai *et al.* [17] conduct a segmentation task through cube map projection, which relieves the distortion near the poles by the top/bottom/left/right/front/back views of the sphere.

To deal with the discontinuity issue in equirectangular projected image, Yang *et al.* [27] conduct object detection task on panoramic images using multi-angle projection. In this approach, the omnidirectional image is represented by multiple overlapping 2D images obtained by re-projecting the equirectangular image from multiple angles among the equator. The overlapping images ensures each point on the sphere will have a neighborhood that lies within at least one projected image, that is, the points of discontinuity that correspond to the edge of any projected image lie within the continuity region of some

other projected image. As such the discontinuity issue is deal with. However, since multiple projected images are required to represent an omnidirectional image, the multi-angle projection requires lots of repeating calculation on each of the multiple projected images in both training and testing phases.



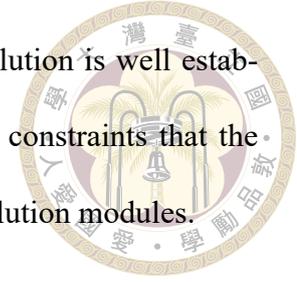
## 2.2 Spherical CNNs

Instead of processing the projected spherical signals on a plane, another approach is to reconstruct the signals on a sphere in essence. Just as the 2D convolution has the shift-invariance property, it is desirable that the convolution on the spherical signals should be rotation-invariant.

Towards this end, in Spherical CNN[3] a rotation invariant convolution on spherical signals was proposed. As an extension, Cohen *et al.* further proposed gauge equivariant convolutional networks[14] for signals on arbitrary manifolds, as well as Icosahedral CNN[2] which is a special case of gauge equivariant convolutional network on spherical signals in the form of icosahedral grids. In gauge equivariant convolution, the feature vectors of neighboring pixels in the receptive field are first transformed to a common local chart through parallel transport[25], at which the convolution is taken as a weighted sum of the corresponding transformed features. This allows for geometric features to be processed in an equivariant manner independent of the particular gauge selection.

In Esteves *et al.*'s work [9], the spherical convolution is computed through spherical fourier transform. That is, the spherical Fourier coefficients of the convoluted signal is computed as the product of the spherical Fourier coefficients of the original signal and the filter.

Although the mathematical theory for rotation-invariant convolution is well established, it either poses major computational overhead or significant constraints that the model should be linear [15], hindering its application to deep convolution modules.



## 2.3 Self-defined Representations and Kernels

There are various representations to a spherical signal, and for each representation various self-defined kernels were proposed to handle the distortion and discontinuity issues. Coor *et al.* [4] and Fernandez *et al.* [10] used equirectangular projected images as input, where the sampling locations of the convolution filters were adapted based on the geometry of the spherical image representation.

Dai *et al.* [6] handled the issue of distorted features by using deformable kernels. The additional parameters, x-, y-offsets, are trained along with the weights of convolution kernels, and deformable kernels can self-adjust the receptive fields by moving the position of input pixels through offsets.

Lee *et al.* [19] addressed the distortion issue through icosahedron representation of spherical signals, where each pixel is a subdivision surface on the icosahedron mesh. To deal with the discontinuity issue, an extra table storing the adjacency information between pixels is maintained for computing convolution and pooling.

In Chao *et al.*'s work [29], the spherical signal is represented as triangular grids through subdivision on icosahedron, and each triangular grid is further represented by regular square grids while dealing the discontinuity issue through padding. A  $3 \times 3$  convolution kernel pretrainable from plain images was then proposed as an arc-based interpolation of the neighboring square grids to mimic convolution on icosahedron, which is a

weighted sum of six neighboring triangular facets of a vertex.

Spherical signals can be sampled more properly on self-defined representations than on planar images. However, the transformation of each self-defined representation requires extra computational cost, and the kernels can not adapt to CNNs originally designed for images with square grids.





## Chapter 3 Method

We consider an equirectangular image by two parts: i) the position of a pixel is regarded as positional information, and ii) the context and the value of channels are regarded as features. In equirectangular projection:

$$f(\lambda, \varphi) = (R(\lambda - \lambda_0), R(\varphi - \varphi_0)). \quad (3.1)$$

Here,  $R$  is the radius of the globe,  $(\lambda, \varphi)$  is the longitude and latitude of the location to project,  $(\lambda_0, \varphi_0)$  is the central parallel and the meridian of the map, and  $(x, y)$  is the coordinate on the equirectangular projected image. If we acquire the coordinate of input pixel we can have the positional information of pixel from eq. 3.1.

Coming up with the positional encoding in self-attention mechanism, we introduce the relationship between convolution and self-attention in sec. 3.1 and positional encoding in sec. 3.2. We propose our structure in sec. 3.5.



### 3.1 Convolution and Self-Attention on Feature Extraction

In 2D convolution the pixel value on position  $(i, j)$  is evaluated as:

$$f(i, j) = \sum_{\substack{\Delta_W \in \{-\lfloor \frac{K}{2} \rfloor, \dots, \lfloor \frac{K}{2} \rfloor\} \\ \Delta_H \in \{-\lfloor \frac{K}{2} \rfloor, \dots, \lfloor \frac{K}{2} \rfloor\}}} F_{(\Delta_W, \Delta_H)} \cdot \mathcal{I}_{(i+\Delta_W, j+\Delta_H)}. \quad (3.2)$$

Here  $F \in \mathbb{R}^{K \times K \times D_{in}}$  is the weights of the convolution filter,  $K$  is the size of the filter, and  $\mathcal{I} \in \mathbb{R}^{W \times H \times D_{in}}$  is the input image of size  $W \times H$  and  $D_{in}$  channels.  $\mathcal{I}_{(i, j)} \in \mathbb{R}^{D_{in}}$  is the channel values of the pixel on position  $(i, j)$ . Note that convolution is a shift-invariant operation, which is not the case for equirectangular projected images. For instance, shifting an object from the equator to the North or South pole on sphere leads to not only a shift in the equirectangular projected image, but also a distortion that depends on latitude. Therefore, a position-dependent filter is needed in panoramic image processing.

Another useful method is self-attention mechanism for computer vision. Let  $I \in \mathbb{R}^{WH \times D_{in}}$  be the flattened image. In self-attention, the output of a query pixel  $q$  is computed as the weighted sum of every key pixel with the weight of attention probabilities, where the attention probabilities measures the similarity between the query pixel and a key pixel as follows:

$$\begin{aligned} \text{Self-Attention}(I)_q &= \sum_k^{WH} \text{softmax}(A_{q,:})_k I_{k,:} W_{val} \\ \text{softmax}(A_{q,:})_k &= \frac{\exp(A_{q,k})}{\sum_p \exp(A_{q,p})}. \end{aligned} \quad (3.3)$$

Here,  $W_{val} \in \mathbb{R}^{D_{in} \times D_{out}}$  linearly transforms the input image  $I$  from  $D_{in}$  channels to  $D_{out}$

channels,  $A \in \mathbb{R}^{W \times H \times W \times H}$  denotes the pairwise attention score between each pixel in the image  $I$ , and  $\text{softmax}(A_{q,:})_k$  is the attention probabilities on pixel  $q$  contributed by pixel  $k$ . The attention score between pixel  $q$  and pixel  $k$  is commonly calculated in an inner product form:

$$A = (IW_{qry})(IW_{key})^T = IW_{qry}W_{key}^T I^T. \quad (3.4)$$

Here,  $W_{qry}, W_{key} \in \mathbb{R}^{D_{in} \times D_{out}}$  linearly transform each pixel from  $\mathbb{R}^{D_{in}}$  to  $\mathbb{R}^{D_{out}}$ , on which the inner product is computed as attention score between pixels. In self-attention mechanism, we can use positional encoding to preserve the positional information of each pixel.

$$A = (I + E)W_{qry}W_{key}^T(I + E)^T, \quad (3.5)$$

where  $E \in \mathbb{R}^{W \times H \times D_{in}}$  is the positional encoding of the pixels. (see Sec. 3.2 for more details)

It has been pointed out by Cordonnier *et al.* [5] that the convolution layer can in fact be realized through multi-head self-attention mechanism, where the position information of each key pixel is taken into account through positional encoding. In this work, we extend the idea of realizing the convolution operation through self-attention mechanism to omnidirectional images. Moreover, we propose **Spherical Encoding** for equirectangular projected image to preserve positional information of each pixel based on the great circle distance. We will further elaborate spherical encoding in sec. 3.2.

## 3.2 Encoding with Positional Information

There are several ways of positional encoding to preserve the positional information of pixels, including *absolute* and *relative* encoding [7]. In *absolute* encoding, each pixel  $p$

is represented by a fixed or learned vector  $E_p^{abs} \in \mathbb{R}^{D_{in}}$ , by which the positional information between a pair of pixels is represented by the dot product between their encoding vectors.

More precisely, the attention score between query pixel and key pixel is computed as:

$$\begin{aligned} A_{(q,k)}^{abs} &= (I_q + E_q^{abs}) W_{qry} W_{key}^T (I_k + E_k^{abs})^T \\ &= I_q W_{qry} W_{key}^T I_k^T + E_q^{abs} W_{qry} W_{key}^T I_k^T + I_q W_{qry} W_{key}^T E_k^{absT} + E_q^{abs} W_{qry} W_{key}^T E_k^{absT}, \end{aligned} \quad (3.6)$$

where  $I_q \in \mathbb{R}^{D_{in}}$  is the  $q^{th}$  pixel in the flattened image  $I$ .

On the contrary, in *relative* encoding, it is the relative position between the key and query pixel  $\mathbf{q}, \mathbf{k} \in [1, W-1] \times [0, H-1]$  that is considered, by which the attention score is computed as:

$$\begin{aligned} A_{\mathbf{k}-\mathbf{q}}^{rel} &= (I_q + e) W_{qry} W_{key}^T (I_k + E_{\mathbf{k}-\mathbf{q}}^{rel})^T \\ &= I_q W_{qry} W_{key}^T I_k^T + I_q W_{qry} (W_{key}^T E_{\mathbf{k}-\mathbf{q}}^{relT}) + (e W_{qry}) W_{key}^T I_k^T + (e W_{qry}) (W_{key}^T E_{\mathbf{k}-\mathbf{q}}^{relT}) \\ &= I_q W_{qry} W_{key}^T I_k^T + I_q W_{qry} W_{key}^T E_{\mathbf{k}-\mathbf{q}}^{relT} + u W_{key}^T I_k^T + v W_{key}^T E_{\mathbf{k}-\mathbf{q}}^{relT}, \end{aligned} \quad (3.7)$$

where  $E_{\mathbf{k}-\mathbf{q}}^{rel} \in \mathbb{R}^{D_{in}}$  is the relative encoding vector that depends on the relative position between the key and query pixels, and  $e \in \mathbb{R}^{D_{in}}$  is a learned or fixed vector.

### 3.3 Spherical absolute encoding

In order to contain the spherical topological information, we propose **Spherical Encoding** based on the great circle distance. For two point  $(\lambda_1, \varphi_1), (\lambda_2, \varphi_2)$  on the unit sphere

in spherical coordinate, the angle in between can be written as:

$$\begin{aligned}
 d((\lambda_1, \varphi_1), (\lambda_2, \varphi_2)) &= \arccos((\cos \lambda_1 \cos \varphi_1, \sin \lambda_1 \cos \varphi_1, \sin \varphi_1) \cdot (\cos \lambda_2 \cos \varphi_2, \sin \lambda_2 \cos \varphi_2, \sin \varphi_2)) \\
 &= \arccos(\sin \varphi_1 \sin \varphi_2 + \cos \varphi_1 \cos \varphi_2 \cos \lambda_1 \cos \lambda_2 + \cos \varphi_1 \cos \varphi_2 \sin \lambda_1 \sin \lambda_2).
 \end{aligned}
 \tag{3.8}$$

Here, the arc-cosine function can be approximated by the Taylor series expansion:

$$\begin{aligned}
 d((\lambda_1, \varphi_1), (\lambda_2, \varphi_2)) &= \frac{\pi}{2} - \sum_{k=0}^{\infty} \frac{1}{2^{2k}} \frac{(2k)!}{(k!)^2} \frac{1}{2k+1} (\sin \varphi_1 \sin \varphi_2 + \cos \varphi_1 \cos \varphi_2 \cos \lambda_1 \cos \lambda_2 + \cos \varphi_1 \cos \varphi_2 \sin \lambda_1 \sin \lambda_2) \\
 &\sim \frac{\pi}{2} - \Psi_n(\lambda_1, \varphi_1) \cdot \Psi_n(\lambda_2, \varphi_2),
 \end{aligned}
 \tag{3.9}$$

Where  $\Psi_n(\lambda, \varphi)$  is the spherical encoding that corresponds to the  $n^{th}$  order Taylor series approximation, which is a  $\sum_{k=0}^n \binom{2k+3}{2} = \frac{2}{3}n^3 + \frac{7}{2}n^2 + \frac{35}{6}n + 3$  dimensional vector where each element (indexed by  $k, p, q, r$  where  $0 \leq k \leq n$ ,  $0 \leq p, q, r$ , and  $p+q+r=2k+1$ ) takes the form:

$$[\Psi_n(\lambda, \varphi)]_{k,p,q,r} = \frac{1}{2^k} \frac{(2k)!}{k!} \sqrt{\frac{1}{p!q!r!}} \sin^p \varphi \cos^{q+r} \varphi \cos^q \lambda \sin^r \lambda,
 \tag{3.10}$$

Though Taylor expansion has infinite terms, as convolution usually operates on local patterns, the low order terms are fairly enough to elaborate positional information of adjacent pixels appropriately. The first order spherical encoding is given by  $\Psi_0(\lambda, \varphi) = (\sin \varphi, \cos \varphi \cos \lambda, \cos \varphi \sin \lambda)$ .

### 3.4 Spherical relative encoding

By rewriting eq.3.9 with  $(\varphi_1, \lambda_1)$  and  $(\varphi_2, \lambda_2)$  replaced by  $(\varphi, \lambda)$  and  $(\varphi + \Delta\varphi, \lambda + \Delta\lambda)$ , respectively, we can express the great circle distance in terms of spherical relative



encoding as follows:

$$\begin{aligned}
 & d((\lambda_1, \varphi_1), (\lambda_2, \varphi_2)) \\
 &= \frac{\pi}{2} - \sum_{k=0}^{\infty} \frac{1}{2^{2k}} \frac{(2k)!}{(k!)^2} \frac{1}{2k+1} (\sin \varphi_1 \sin \varphi_2 + \cos \varphi_1 \cos \varphi_2 \cos \lambda_1 \cos \lambda_2 + \cos \varphi_1 \cos \varphi_2 \sin \lambda_1 \sin \lambda_2)^{2k+1} \\
 &= \frac{\pi}{2} - \sum_{k=0}^{\infty} \frac{1}{2^{2k}} \frac{(2k)!}{(k!)^2} \frac{1}{2k+1} (\sin^2 \varphi \cos \Delta \varphi + \sin \varphi \cos \varphi \sin \Delta \varphi + \cos^2 \varphi \cos \Delta \varphi \cos \Delta \lambda - \cos \varphi \sin \varphi \sin \Delta \varphi \cos \Delta \lambda)^{2k+1} \\
 &\approx \frac{\pi}{2} - \Phi_n^{(qry)}(\varphi, \lambda)^T \Phi_n^{(key)}(\Delta \varphi, \Delta \lambda)
 \end{aligned} \tag{3.11}$$

Here  $\Phi_n^{(qry)}(\varphi, \lambda)$  and  $\Phi_n^{(key)}(\Delta \varphi, \Delta \lambda)$  are the query/key encodings that correspond to the  $n^{th}$  order Taylor series approximation, respectively. The query encoding  $\Phi_n^{(qry)}(\varphi, \lambda)$  is a  $\frac{1}{3}k^4 + \frac{8}{3}k^3 + \frac{23}{3}k^2 + \frac{28}{3}k + 4$  dimensional vector, with each element (indexed by  $k, p, q, r, s$  where  $0 \leq k \leq n$ ,  $p, q, r, s \geq 0$ , and  $p+q+r+s=2k+1$ ) taking the form

$$[\Phi_n^{(qry)}(\varphi, \lambda)]_{k,p,q,r,s} = \frac{1}{2^k} \frac{(2k)!}{k!} \sqrt{\frac{1}{p!q!r!s!}} (-1)^s \sin^{2p+q+s} \varphi \cos^{q+2r+s} \varphi.$$

The key encoding has the same dimension with elements taking the form

$$[\Phi_n^{(key)}(\Delta \varphi, \Delta \lambda)]_{k,p,q,r,s} = \frac{1}{2^k} \frac{(2k)!}{k!} \sqrt{\frac{1}{p!q!r!s!}} \cos^{p+r} \Delta \varphi \sin^{q+s} \Delta \varphi \cos^{r+s} \Delta \lambda$$

In particular, for first order approximation the query and key encodings are given by

$$\begin{aligned}
 \Phi_0^{(query)}(\varphi, \lambda) &= (\sin^2 \varphi, \sin \varphi \cos \varphi, \cos^2 \varphi, -\sin \varphi \cos \varphi) \\
 \Phi_0^{(key)}(\Delta \varphi, \Delta \lambda) &= (\cos \Delta \varphi, \sin \Delta \varphi, \cos \Delta \varphi \cos \Delta \lambda, \sin \Delta \varphi \cos \Delta \lambda)
 \end{aligned}$$

Spherical relative encoding allows the proximity information between query and key pixels on the sphere to be represented as two integral parts in equirectangular format: the query encoding that only depends on the location of the query pixel  $(\varphi, \lambda)$ , and the key encoding that solely depends on the relative position  $(\Delta \varphi, \Delta \lambda)$ . This allows us to effec-

tively adjust the attention scores for pixels in the receptive field of equirectangular format based on its proximity on the sphere.



### 3.5 Encoding and Convolution

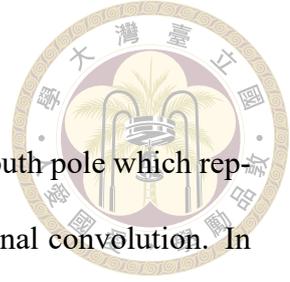
Though self-attention model hits a great success on various computer vision[24][30] and natural language processing[8] applications, its dire memory usage poses a serious issue to be reckoned with. More precisely, for an input image of size  $W \times H$  and  $D_{in} \in O(WH)$  channels, the self-attention operation requires  $O(W^2H^2)$  memory usage, a  $\frac{WH}{D_{in}+D_{out}}$ -fold increase compared to the convolution-based approach which requires  $O(WH(D_{in}+D_{out}))$  memory usage.

To relieve the huge memory usage of self-attention, a common approach is convolution-based attention[28][22] which uses additional parameters and incorporates techniques such as spatial attention or channel attention to learn the importance of features. This leads to memory usage in the order of  $O(WH \times (D_{in}+D_{out}))$ , for which  $D_{in}$  and  $D_{out}$  are usually much smaller than  $WH$ .

Instead of using additional parameters, here we use the inner product of spherical encoding as the attention map to refine the convolution features:

$$\begin{aligned}
 f^*(i,j) &\sim \sum_{\substack{\Delta_W \in \{-\lfloor \frac{K}{2} \rfloor \dots \lfloor \frac{K}{2} \rfloor\} \\ \Delta_H \in \{-\lfloor \frac{K}{2} \rfloor \dots \lfloor \frac{K}{2} \rfloor\}}} \frac{e^{d((\lambda_i, \varphi_j), (\lambda_{i+\Delta_W}, \varphi_{j+\Delta_H}))}}{\sum_{\substack{\Delta'_W \in \{-\lfloor \frac{K}{2} \rfloor \dots \lfloor \frac{K}{2} \rfloor\} \\ \Delta'_H \in \{-\lfloor \frac{K}{2} \rfloor \dots \lfloor \frac{K}{2} \rfloor\}}} e^{d((\lambda_i, \varphi_j), (\lambda_{i+\Delta'_W}, \varphi_{j+\Delta'_H}))}} F_{(\Delta_W, \Delta_H)} \cdot \mathfrak{J}(i+\Delta_W, j+\Delta_H) \\
 &= \sum_{\substack{\Delta_W \in \{-\lfloor \frac{K}{2} \rfloor \dots \lfloor \frac{K}{2} \rfloor\} \\ \Delta_H \in \{-\lfloor \frac{K}{2} \rfloor \dots \lfloor \frac{K}{2} \rfloor\}}} \frac{e^{-\Psi_n(\lambda_i, \varphi_j) \cdot \Psi_n(\lambda_{i+\Delta_W}, \varphi_{j+\Delta_H})}}{\sum_{\substack{\Delta'_W \in \{-\lfloor \frac{K}{2} \rfloor \dots \lfloor \frac{K}{2} \rfloor\} \\ \Delta'_H \in \{-\lfloor \frac{K}{2} \rfloor \dots \lfloor \frac{K}{2} \rfloor\}}} e^{-\Psi_n(\lambda_i, \varphi_j) \cdot \Psi_n(\lambda_{i+\Delta'_W}, \varphi_{j+\Delta'_H})}} F_{(\Delta_W, \Delta_H)} \cdot \mathfrak{J}(i+\Delta_W, j+\Delta_H),
 \end{aligned} \tag{3.12}$$

where  $(\lambda_i, \varphi_j)$  is the longitude and latitude of the pixel  $(i, j)$  on the equirectangular pro-



jected image.

In equirectangular projected images, pixels near the north and south pole which represent small area on the sphere will be overly weighted in conventional convolution. In our proposed spherical encoding, however, observe that in the most prevailing  $3 \times 3$  kernel size scenario, the great circle distance between center pixel and high-latitude pixels are generally smaller than that of low-latitude pixels (fig. 3.1). This generally leads to smaller weights assigned to high-latitude pixels and mitigates the issue of overly weighted high-latitude pixels suffered in conventional convolution on equirectangular projected images. Similarly, the discontinuity of two sides of the image can also be fixed by the  $\sin\varphi$  term in the spherical encoding.

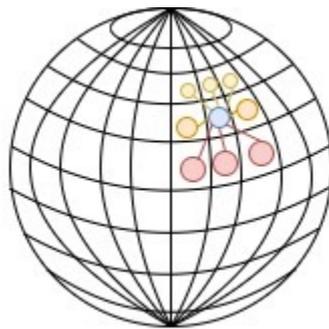


Figure 3.1: In the  $3 \times 3$  kernel size setting, the pixels which are further away from the center (blue) represent greater surface area, and is assigned more attention score through **Spherical encoding**. (eq. 3.12)

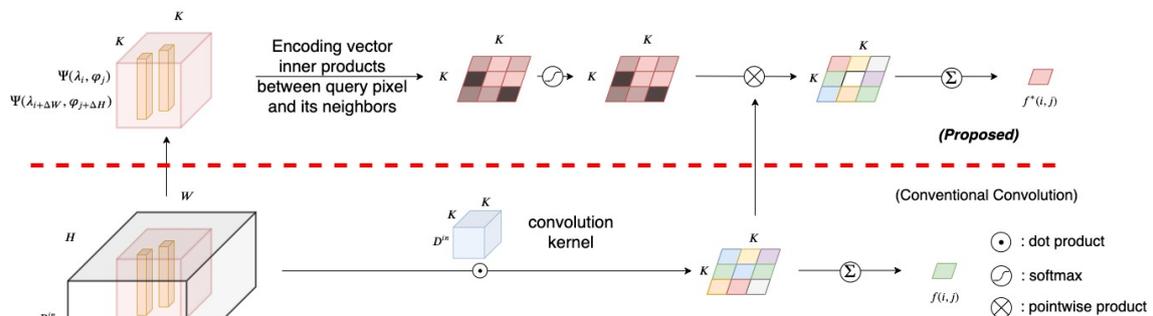


Figure 3.2: Process of **Spherical encoding** (eq. 3.12). Input can be either omnidirectional images or feature maps from previous layers.



## Chapter 4 Experiment

### 4.1 Dataset

Omnidirectional MNIST is proposed by Coors *et al.*[4]. The raw MNIST[18] image is placed on tangent planes randomly drawn from the sphere, and the equirectangular projection of the scene is rendered at a resolution of 60 x 60. Similar method can be applied to generate the omnidirectional datasets omniCIFAR from CIFAR10 and CIFAR100[16].



Figure 4.1: Omnidirectional dataset includes images with severe distortion around the North and South poles and discontinuity on the left and right sides.

### 4.2 Classification on omniMNIST

Due to a variety of representations of spherical signals, we conduct the experiments on the proposed spherical encoding as well as a variety of baselines mentioned in [4]. Here S2CNN[3] is a convolution architecture defined in the  $SO(3)$  group; GCNN[13] takes a spherical signal as a weighted graph and the weights of the edges are given from great

circle distance; SpherePHD[19] represents the spherical signals on icosahedral mesh, and proposes convolution and pooling under this representation.



### 4.2.1 Experiment Setup

We followed the experiment setup in [4], and used omniMNIST as benchmark to compare the feature extraction methods. The backbone network is composed of two convolution layers and max-pooling, followed by a fully-connected layer with 10 outputs. The first convolution layer has 32 filters with kernel size  $3 \times 3$ , and the second convolution layer has 64 filters with kernel size  $3 \times 3$ . Each convolution layer is followed by a ReLU activation function, and the fully connected layer is followed by a softmax function. For multi-layered model, we added more convolution layers with 64 filters, where the results are reported in Table 4.2.

Method	Input Type	Accuracy
Conventional Convolution	Cubemap Image	0.8997
Conventional Convolution	Equirectangular Image	0.9039
SpherePHD [19]	Octahedral Representations	0.8813
S2CNN[3]	$SO(3)$ signal	0.8814
GCNN[13]	Graph	0.8279
SphereNet[4]	Equirectangular Image	0.9402
Spherical Encoding ( <i>ours</i> )	Equirectangular Image	0.9322

Table 4.1: Classification accuracy comparison of the proposed spherical encoding and various baselines on omni-MNIST.

Convolution	Accuracy				
	2 layer	3 layer	4 layer	5 layer	ResNet18
SphereNet	0.9402	0.9718	0.9740	0.9743	0.9910
Spherical Encoding ( <i>ours</i> )	0.9322	0.9513	0.9666	0.9720	0.9915

Table 4.2: Classification accuracy on multi-layered models of various depths on omni-MNIST.



## 4.2.2 Results

Table 4.1 compares the accuracy performance of the proposed spherical encoding and various baselines on omniMNIST. It follows that the proposed spherical encoding has better ability to extract spherical features on equirectangular projected images and outperforms conventional convolution.

Though the proposed spherical encoding slightly falls behind SphereNet on the shallow two-layer CNN backbone, our method can be easily adopted to deeper models. Table. 4.2 shows that the proposed spherical encoding is capable of adapting to various existing deep learning model and the nowadays widely adopted residual modules such as ResNet18[11], achieving the state-of-the-art performance on omniMNIST classification task.

## 4.3 Residual Module

Residual module has become a common architecture design for deep learning which often makes deep network structure more easily to optimize, and achieves better performance[11]. To elaborate the adaptability of the proposed spherical encoding in deeper models with residual modules, we conduct experiments on classification task over omnidirectional datasets, and demonstrate the ability of feature extraction in terms of classification accuracy.



### 4.3.1 Experiment Setup

The Resnet18 (fig. 4.2) is taken as the backbone where the convolution unit is replaced by (a) conventional convolution, (b) SphereNet convolution, and (c) spherical encoding with convolution (*our method*). Note that in the experiment of SphereNet convolution, the pooling functions are substituted by the spherical pooling in their own work [4], for the purpose of addressing the issue of oversampling on high latitude regions.

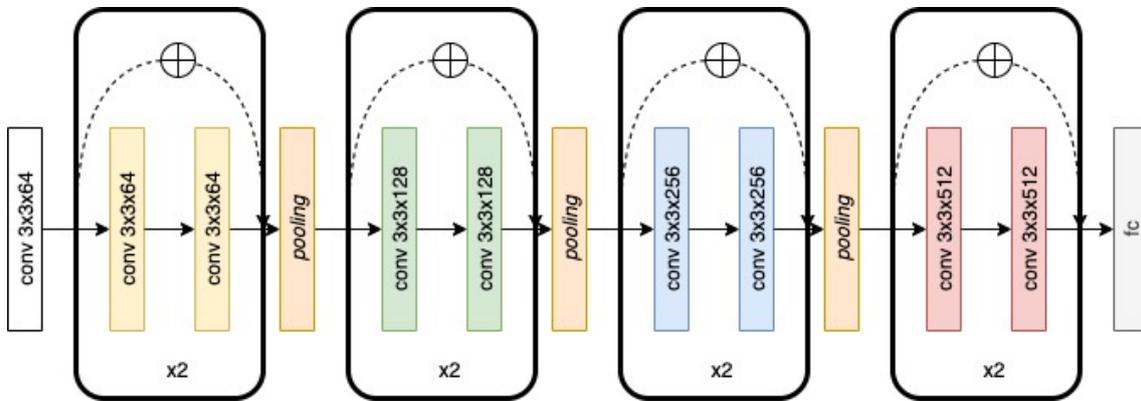


Figure 4.2: The Resnet-18 backbone.

Convolution	Dataset	Accuracy
Conventional Convolution	omni-CIFAR10	0.8339
SphereNet Convolution	omni-CIFAR10	0.8346
Spherical Encoding(ours)	omni-CIFAR10	<b>0.8461</b>
Conventional Convolution	omni-CIFAR100	0.6020
SphereNet Convolution	omni-CIFAR100	0.5979
Spherical Encoding(ours)	omni-CIFAR100	<b>0.6413</b>

Table 4.3: Comparison of various convolution/encoding schemes on ResNet18.

### 4.3.2 Result

As demonstrated in Table. 4.3, the proposed spherical encoding yields the best results. It is also observed that SphereNet convolution performs slightly worse than conventional convolution in the omniCIFAR100 classification task. This indicates that SphereNet

convolution does not benefit from the residual structures. Here we give an intuitive explanation as follows (as shown in Table.4.4):



As illustrated in fig. 4.3, in SphereNet convolution the distortion issue is dealt with by maintaining the size of receptive field on the tangent plane, with the output representing the convolution of spherical data on the tangent plane. This leads to a receptive field whose size depends on the latitude in the equirectangular format. As in residue module the convolution result is added to the original input in equirectangular format, we hypothesize that the latitude-varying receptive field in the equirectangular format causes inconsistency in the addition operation in residual modules.

In contrast to SphereNet, in our method the size of receptive field remains identical in the equirectangular format, regardless of the latitude. That is, the convolution is computed as a weighted sum of samples drawn from a surface region whose latitude and longitude spans  $K\Delta\varphi$  and  $K\Delta\lambda$ , respectively. Here  $K$  denotes the kernel size, while  $\Delta\varphi$  and  $\Delta\lambda$  denotes the latitude and longitude spans of the surface region that a pixel in equirectangular format represents, respectively. The distortion issue is instead dealt with through spherical encoding, where pixels near the North and South poles will be assigned smaller weights computed as the dot product of spherical encoding, which is consistent to the intuition that high latitude pixels in equirectangular format represents smaller surface regions on the sphere.

## 4.4 Self-Attention Model

We followed the experiment setup in [5] and [24] to demonstrate the effect of various encoding schemes, where conventional convolution and fully self-attention based models

	operation	input	physical meaning
Conventional Convolution	convolution	adjacent pixels	weighted sum in local region
SphereNet Convolution[4]	convolution	pixels calculated from gnomonic projection	weighted sum on tangent plane
Spherical Encoding (ours)	convolution with weight from encoding	adjacent pixels	weighted sum in local region with smaller weight for higher latitude pixels (eq. 3.9)

Table 4.4: Comparison between spherical encoding, convolution, and SphereNet

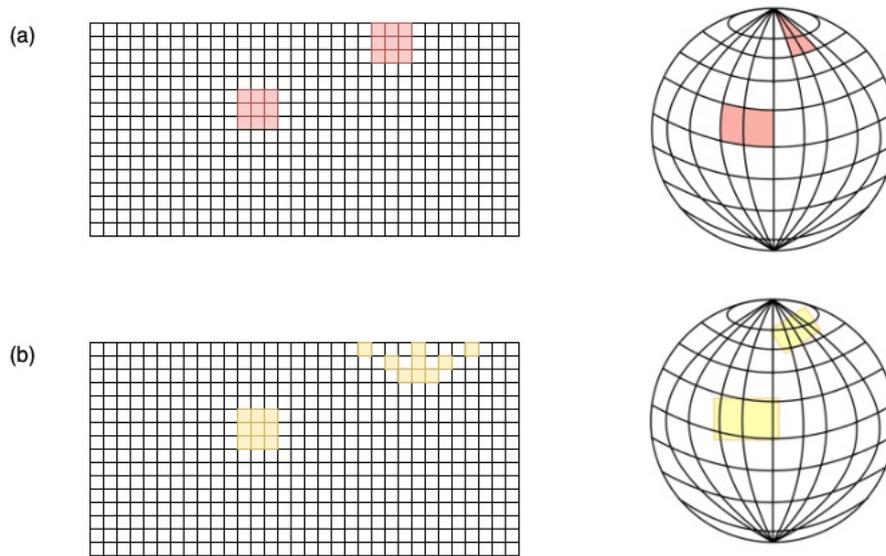


Figure 4.3: The physical meaning of pixel on equirectangular projection and on SphereNet kernel. (a) The pixel on the actual sphere surface by equirectangular projection (b) The receptive field of SphereNet kernel.

are compared over omnidirectional image datasets. We use the aforementioned Resnet18 as the backbone, and replaced the convolution layers with self-attention layers incorporated with various encoding schemes, as illustrated in Table. 4.5 and 4.6.

#### 4.4.1 Result

As illustrated in Table. 4.5 and 4.6, the fully self-attention model with either absolute or relative encoding does not perform as good as conventional convolutions for spherical data. For fully self-attention models, the proposed absolute spherical encoding yields bet-



Omni-CIFAR10		
Feature extractor	Encoding	Accuracy
Conventional Convolution	-	0.8339
Self-Attention	absolute encoding	0.8005
Self-Attention	relative encoding	0.8178
Self-Attention	spherical encoding (abs 0 <sup>th</sup> order)	0.8230
Self-Attention	spherical encoding (rel 0 <sup>th</sup> order)	0.8000
Conventional Convolution	spherical encoding (abs 0 <sup>th</sup> order)	0.8461
Conventional Convolution	spherical encoding (abs 1 <sup>st</sup> order)	<b>0.8511</b>
Conventional Convolution	spherical encoding (abs 2 <sup>nd</sup> order)	0.8369
Conventional Convolution	spherical encoding (rel 0 <sup>th</sup> order)	0.8331
Conventional Convolution	spherical encoding (rel 1 <sup>st</sup> order)	0.8389

Table 4.5: results for different encodings on omni-cifar10

Omni-CIFAR100		
Feature extractor	Encoding	Accuracy
Conventional Convolution	-	0.6020
Self-Attention	absolute encoding	0.5747
Self-Attention	relative encoding	0.5856
Self-Attention	spherical encoding (abs 0 <sup>th</sup> order)	0.5912
Self-Attention	spherical encoding (rel 0 <sup>th</sup> order)	0.5815
Conventional Convolution	spherical encoding (abs 0 <sup>th</sup> order)	0.6404
Conventional Convolution	spherical encoding (abs 1 <sup>st</sup> order)	<b>0.6413</b>
Conventional Convolution	spherical encoding (abs 2 <sup>nd</sup> order)	0.6228
Conventional Convolution	spherical encoding (rel 0 <sup>th</sup> order)	0.6186
Conventional Convolution	spherical encoding (rel 1 <sup>st</sup> order)	0.6246

Table 4.6: Comparison of various encoding schemes on omni-cifar100

ter result over traditional absolute and relative encoding. The conventional convolution also benefits from absolute spherical encoding. The relationship between the surface area where each pixel stands and the great circle distance between pixels is data-driven. The order of approximation can be used as a hyperparameter in the training phase, and further improvements can be achieved by tuning the order of approximation in spherical encoding.





## Chapter 5 Conclusion

In this work, we propose **Spherical Encoding** for omnidirectional images, and compare it with self-attention models and convolution models on omnidirectional image dataset. Spherical encoding preserves spatial information on the sphere, and can be easily adapted to both convolution and self-attention schemes in deep learning models. Experiments show that both conventional convolution and self-attention models benefit from spherical encoding on classification tasks. For deeper models, spherical encoding can be integrated with residual module, leading to state-of-the-art performance.

As future work, we will further adapt spherical encoding to various other deep learning models as well as omnidirectional image related tasks. We will also explore spherical encoding defined over self-defined distances other than the great circle distance discussed in this work.

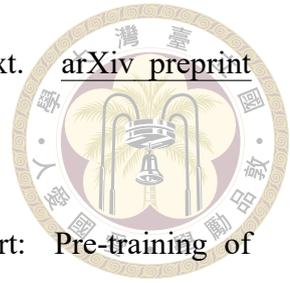


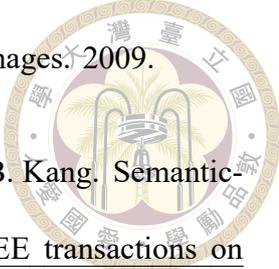


## References

- [1] Faq.md. <https://github.com/jonas-koehler/s2cnn/blob/master/FAQ.md>.
- [2] T. Cohen, M. Weiler, B. Kicanaoglu, and M. Welling. Gauge equivariant convolutional networks and the icosahedral cnn. In International Conference on Machine Learning, pages 1321–1330. PMLR, 2019.
- [3] T. S. Cohen, M. Geiger, J. Köhler, and M. Welling. Spherical cnns. arXiv preprint arXiv:1801.10130, 2018.
- [4] B. Coors, A. P. Condurache, and A. Geiger. Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In Proceedings of the European Conference on Computer Vision (ECCV), pages 518–533, 2018.
- [5] J.-B. Cordonnier, A. Loukas, and M. Jaggi. On the relationship between self-attention and convolutional layers. arXiv preprint arXiv:1911.03584, 2019.
- [6] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In Proceedings of the IEEE international conference on computer vision, pages 764–773, 2017.
- [7] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov. Transformer-

- xl: Attentive language models beyond a fixed-length context. [arXiv preprint arXiv:1901.02860](#), 2019.
- [8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. [arXiv preprint arXiv:1810.04805](#), 2018.
- [9] C. Esteves, C. Allen-Blanchette, A. Makadia, and K. Daniilidis. Learning so(3) equivariant representations with spherical cnns. In ECCV, 2018.
- [10] C. Fernandez-Labrador, J. M. Facil, A. Perez-Yus, C. Demonceaux, J. Civera, and J. J. Guerrero. Corners for layout: End-to-end layout recovery from 360 images. IEEE Robotics and Automation Letters, 5(2):1255–1262, 2020.
- [11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.
- [12] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3 d shape descriptors. In Symposium on geometry processing, volume 6, pages 156–164, 2003.
- [13] R. Khasanova and P. Frossard. Graph-based classification of omnidirectional images. In Proceedings of the IEEE International Conference on Computer Vision Workshops, pages 869–878, 2017.
- [14] B. Kicanaoglu, P. de Haan, and T. Cohen. Gauge equivariant spherical cnns. 2019.
- [15] J. Klicpera, J. Groß, and S. Günnemann. Directional message passing for molecular graphs. [arXiv preprint arXiv:2003.03123](#), 2020.



- 
- [16] A. Krizhevsky. Learning multiple layers of features from tiny images. 2009.
- [17] W.-S. Lai, Y. Huang, N. Joshi, C. Buehler, M.-H. Yang, and S. B. Kang. Semantic-driven generation of hyperlapse from 360 degree video. IEEE transactions on visualization and computer graphics, 24(9):2610–2621, 2017.
- [18] Y. LeCun. The mnist database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>, 1998.
- [19] Y. Lee, J. Jeong, J. Yun, W. Cho, and K.-J. Yoon. Spherephd: Applying cnns on a spherical polyhedron representation of 360deg images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9181–9189, 2019.
- [20] Matterport. <https://matterport.com/>.
- [21] D. Maturana and S. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 922–928. IEEE, 2015.
- [22] J. Park, S. Woo, J.-Y. Lee, and I. S. Kweon. Bam: Bottleneck attention module. arXiv preprint arXiv:1807.06514, 2018.
- [23] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 652–660, 2017.
- [24] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens. Stand-alone self-attention in vision models. arXiv preprint arXiv:1906.05909, 2019.

- 
- [25] S. C. Schonsheck, B. Dong, and R. Lai. Parallel transport convolution: A new tool for convolutional neural networks on manifolds. arXiv preprint arXiv:1805.07857, 2018.
- [26] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1912–1920, 2015.
- [27] W. Yang, Y. Qian, J.-K. Kämäräinen, F. Cricri, and L. Fan. Object detection in equirectangular panorama. In 2018 24th International Conference on Pattern Recognition (ICPR), pages 2190–2195. IEEE, 2018.
- [28] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Free-form image inpainting with gated convolution. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 4471–4480, 2019.
- [29] C. Zhang, S. Liwicki, W. Smith, and R. Cipolla. Orientation-aware semantic segmentation on icosahedron spheres. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 3533–3541, 2019.
- [30] H. Zhao, J. Jia, and V. Koltun. Exploring self-attention for image recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10076–10085, 2020.